



A
PROJECT REPORT
ON
COMPUTER HARDWARE SHOP MANAGEMENT SYSTEM

BY
MANE DEEPAK MAHADEV {SEAT NO. 21114}
BHOSALE DARSHAN BHIMRAO {SEAT NO. 21019}
NEVKAR KUNAL NITYANAND {SEAT NO. 21130}
KAMLAPURKAR SHRINIVAS DEEPAK {SEAT NO. 21085}
VARSALE KAPIL RAJENDRA {SEAT NO. 21194}

SAVITRIBAI PHULE
UNIVERSITY OF PUNE

MASTER IN COMPUTER APPLICATION
AUDYOGIK SHIKSHAN MANDAL'S INSTITUTE OF
BUSINESS MANAGEMENT RESEARCH

PUNE-411019

2022-23



SESSION 2022-2023

CERTIFICATE

This is to certify that project entitled “Computer Hardware Shop Management System” has been developed by MCA student of ASM IBMR Department, PUNE towards partial fulfilment of the requirements for the award of the degree of Master of Computer Application is a genuine record of the work carried out by Deepak Mane [21114], Darshan Bhosale [21019], Kunal Nevkar [21130], Shrinivas Kamlapurkar [21085] and Kapil Varsale [21194] under my supervision and guidance and the project report is the original work of student. We have worked on the project Computer Hardware Shop Management System.

Date:

Place: Pune

Signature of Guidance

Dr. Ashwyn Kumar

ACKNOWLEDGEMENT

The completion of this project work could be possible with continued & dedicated efforts & guidance of large number of faculty & staff members of the Institute. We acknowledge our gratitude to all of them. The acknowledgement however will be incomplete without specific mention as follows.

It is my pleasure to acknowledge you that I have received a project on Employee voting system from my teacher.

My first sincere appreciation and gratitude goes to Dr. Ashwyn Kumar for her guidance, constructive comments, valuable suggestions and inspirations. During marking of my project he helped me a lot.

All the work done in coming up with this system is dedicated to my family for being with part of me in the whole process.

CONTENTS

CHAPTER-1 Introduction.....	1
CHAPTER-2 System Analysis.....	2 to 4
2.1 Description of Existing System.....	2
2.2 Drawback of Existing System.....	2
2.3 Proposed System.....	2
2.4 Feasibility Study.....	3
2.4.1 Technical Feasibility.....	3
2.4.2 Economical Feasibility.....	3
2.4.3 Operational Feasibility.....	4
2.5 System Specification.....	4
2.5.1 Software Requirements.....	4
2.5.2 Hardware Requirements.....	4
CHAPTER-3 Overview Of The Language Used.....	5 to 7
3.1 About Java.....	5
3.1.1 Important Java Application.....	5
3.1.2 History of Java Programming Language.....	5
3.2 About MySQL.....	5
3.2.1 RDBMS Terminology.....	6
3.2.2 MySQL Database.....	6 and 7
CHAPTER-4 System Modules.....	8
CHAPTER-5 System Design.....	9 to 15
5.1 Use Case Diagram.....	9
5.2 Sequence Diagram.....	10
5.3 State Diagram.....	11
5.4 Tables.....	12 to 15
CHAPTER-6 System Coding and Form Layout.....	16 to 58
CHAPTER-7 Testing.....	59 to 60
7.1 Testing Approaches.....	59
7.1.1 White Box Testing.....	59
7.1.2 Black Box Testing.....	59
7.2 Unit Testing.....	59
7.3 Integration Testing.....	59
7.4 System Testing (End to End Testing).....	59
7.5 Acceptance Testing.....	60
CHAPTER-8 System Implementation.....	61
CHAPTER-9 Future Enhancement.....	62
CHAPTER-10 Conclusion.....	63
CHAPTER-11 Bibliography.....	64

CHAPTER-1

1.1 INTRODUCTION

This software enables the products acquisition of sales and return operations on stalls and shop-in-shop systems with integrated product sale software and back-office functions. This software controls the entire lifecycle in the store through receiving, sales, inventory management. Role-based security ensures that access to specific functionality is limited to appropriate levels.

The Computer Hardware Shop project is about the working process of Computer store including purchase, sales, stock maintaining, manpower utilization, maintaining customer and supplier information. The project provides computerization of the whole work, thus it helps to improve the quality and productivity of the store. The project will help store employees to maintain all related information about the purchase products, sales product, stock available in the store, supplier and customer. This project decreases the work load of store employees and provides suitable information about each product so that with the help of this project store employees can easily and quickly find the products, supplier, customer information whenever necessary.

The Computer Hardware Shop is to develop software based information of Computer goods shopping. In global business market the electronic products are most important accessories in real life. Visualizing the huge opportunity, this is an effort to maximize the business through the development of this software and keeping the data and thus increasing the customer base from the local as well as global markets around the world.

My ongoing project is a software application because nowadays software is a prominent tool of marketing mantra. With the advent of the software technologies, world has become a global village. Every year, millions more people around the world are added to the existing customer base. So considering a big hike in the revenue in this booming sector and one of the successful businesses through this software one should be proud to have such a technical deal.

CHAPTER-2

SYSTEM ANALYSIS

2.1 DESCRIPTION OF EXISTING SYSTEM

The project on “COMPUTER HARDWARE SHOP” enables to keep the records in the computer system with JAVA as Front-end and Microsoft SQL Server as Backbend.

1. The project keeps the detail record of Suppliers, Customers, Stock of parts, Employees of the Store.
2. The project provides the quotation of product to the customer which enables the customers to make his choice.
3. If the product not available in the store then Shopkeeper order to supplier for providing particular product which customer want.
4. Then related information about that ordered product stored in database of the store.
5. After the order given by the customer it generate the bill.

2.2 Drawback Of Existing System

The existing system follows the same old approach of dealing with the customers in person and storing their records and all other data in files and registers.

This manual system is time consuming and not feasible to use. The existing system is completely manual and depended on human beings.

It has following limitations:

- Data has to be stored manually in files and registers.
- Retrieval of this data is very slow and hectic because one has to search all the files manually.
- As the INTEGER of files increases, it takes up lot of space making it difficult to manage.
- Searching for old bills and files or registers takes a lot of time.
- Data loss is always a major concern as a file or a register can be easily misplaced.
- Natural calamity like floods, fire break-out and other such things can destroy the data, and it can lead to data loss.

2.3 Proposed System

The proposed system that we are making will be completely different and more importantly it will be user friendly.

It can be operated by anyone .It comes with some up-to-date functions which you can perform by just clicking a mouse button.

The aim of this system is to advance to the next level by making everything computerized. It has advantages like this:

- 1) It's user-friendly and it's completely computerized.
- 2) It has a huge data base which can store lots and lots of information.
- 3) Data can be easily entered, updated. I.e. data can be managed easily.
- 4) No need to worry about the storage of the data as it will be stored in your computer.
- 5) Search results can be shown to the clients at the same place and proper and correct information can be provided.

2.4 Feasibility Study

- In preliminary investigation, we examine the feasibility study to understanding the possibility of proposed system being useful in future.
- A feasibility study is a management oriented activity.
- It gives us the cost & benefits of our software define for proposed system.
- The feasibility studied are categories as-

- ✓ **Technical Feasibility**
- ✓ **Economical Feasibility**
- ✓ **Operational Feasibility**

2.4.1 Technical Feasibility

1. A technical feasibility study is an excellent tool for trouble -shooting and long-term planning.
2. Proposed application will provide enough response to enquiries regardless of the INTEGER of users.
3. This software provides lot of security, accuracy & reliability rather than existing system.
4. The proposed new system can easily be expanded to satisfy the new requirements in future.
5. Technical feasibility also acts as flow chat to show how our application evolves & adapts as per requirement.
6. The proposed equipment can safely handle the data required to use the new system.

As per all above point, proposed system is technically feasible.

2.4.2 Economic Feasibility

1. Economic feasibility is a measure of the cost effectiveness of a project or a solution.
2. The cost to implement proposed system includes cost of full systems investigation, cost of hardware and software and the cost of user training.
3. The cost to implement the proposed system is one time investment later they need not spent any more.
4. The organization must get benefit if they invest on the proposed system development.

As per the above reasons the proposed new system is economically feasible.

2.4.3 Operational Feasibility

1. Operational feasibility is mainly concerned with issue like whether the system will be developed or implemented properly.
2. The proposed system will not cause any problem & show efficient performance.
3. This system reduces the work load & work time and leads to error free & accurate work.
4. The proposed system will reduce the space required to store paper work for the previous records.
5. With the help of this software, we get the information about employees, database & other detail easily.
6. From the beginning of proposed system, we made users to involve in some way reducing the resistance in all respects.

As per all above point, proposed system is operationally feasible.

2.5 System Specification

2.5.1 Software Requirements

1. **Front End** :- Java
2. **Back End** :- MySql
3. **Operating System** :- Windows 10 Professional 64-bit operating system, x64-based processor

2.5.2 Hardware Requirements

1. **CPU Type** :- Intel(R) Pentium(R) CPU 3825U
2. **RAM** :- 4.00 GB
3. **CPU speed** :- 1.97 GHz
4. **Display** :- VGA (800*600)
5. **Hard disk** :- 1 TB

CHAPTER-3

OVERVIEW OF THE LANGUAGE USED

3.1 About JAVA

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.

Java Platform is a collection of programs that help programmers to develop and run Java programming applications efficiently. It includes an execution engine, a compiler, and a set of libraries in it. It is a set of computer software and specifications. James Gosling developed the Java platform at Sun Microsystems, and the Oracle Corporation later acquired it.

3.1.1 Here are some important Java applications:

- It is used for developing Android Apps
- Helps you to create Enterprise Software
- Wide range of Mobile java Applications
- Scientific Computing Applications
- Use for Big Data Analytics
- Java Programming of Hardware devices
- Used for Server-Side Technologies like Apache, JBoss, GlassFish, etc.

3.1.2 History of Java Programming Language :

- The Java language was initially called OAK.
- Originally, it was developed for handling portable devices and set-top boxes. Oak was a massive failure.
- In 1995, Sun changed the name to "Java" and modified the language to take advantage of the burgeoning www (World Wide Web) development business.
- Later, in 2009, Oracle Corporation acquired Sun Microsystems and took ownership of three key Sun software assets: Java, MySQL, and Solaris.

3.2 About My SQL

MySQL was created by a Swedish company, MySQL AB, founded by Swedes David Axmark, Allan Larsson and Finland Swede Michael "Monty" Widenius. Original development of MySQL by Widenius and Axmark began in 1994. The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from mSQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as mSQL. By keeping the API consistent with the

mSQL system, many developers were able to use MySQL instead of the (proprietary licensed) mSQL antecedent.

3.2.1 RDBMS Terminology

Before we proceed to explain the MySQL database system, let us revise a few definitions related to the database.

- **Database** – A database is a collection of tables, with related data.
- **Table** – A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column** – One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row** – A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- **Redundancy** – Storing data twice, redundantly to make the system faster.
- **Primary Key** – A primary key is unique. A key value can not occur twice in one table. With a key, you can only find one row.
- **Foreign Key** – A foreign key is the linking pin between two tables.
- **Compound Key** – A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index** – An index in a database resembles an index at the back of a book.
- **Referential Integrity** – Referential Integrity makes sure that a foreign key value always points to an existing row.

3.2.2 MySQL Database

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons –

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

CHAPTER-4

SYSTEM MODULES

Sr. No.	Form Name	Purpose
1	Splash.java	This form display the title of system name of organization & copyright of the system.
2	Login.java	This form is used for security with the help of password.
3	Menuscreen.java	This is the multiple documentation interface for the all of the form in the project.
4	Emlpoyees.java	This form is used to store the employee's information.
5	Customer.java	This form is used to store the customer's information.
6	Supplier.java	This form is used to store the supplier information.
7	MRN.java	This form is used to store the purchase stock entry.
8	Order.java	This form is used to store the order information.
9	Payment.java	This form is used for order payment detail.
10	Bill.java	This form is used for the details of order bill.
11.	Dispatch.java	This form is used to store information about order dispatch.

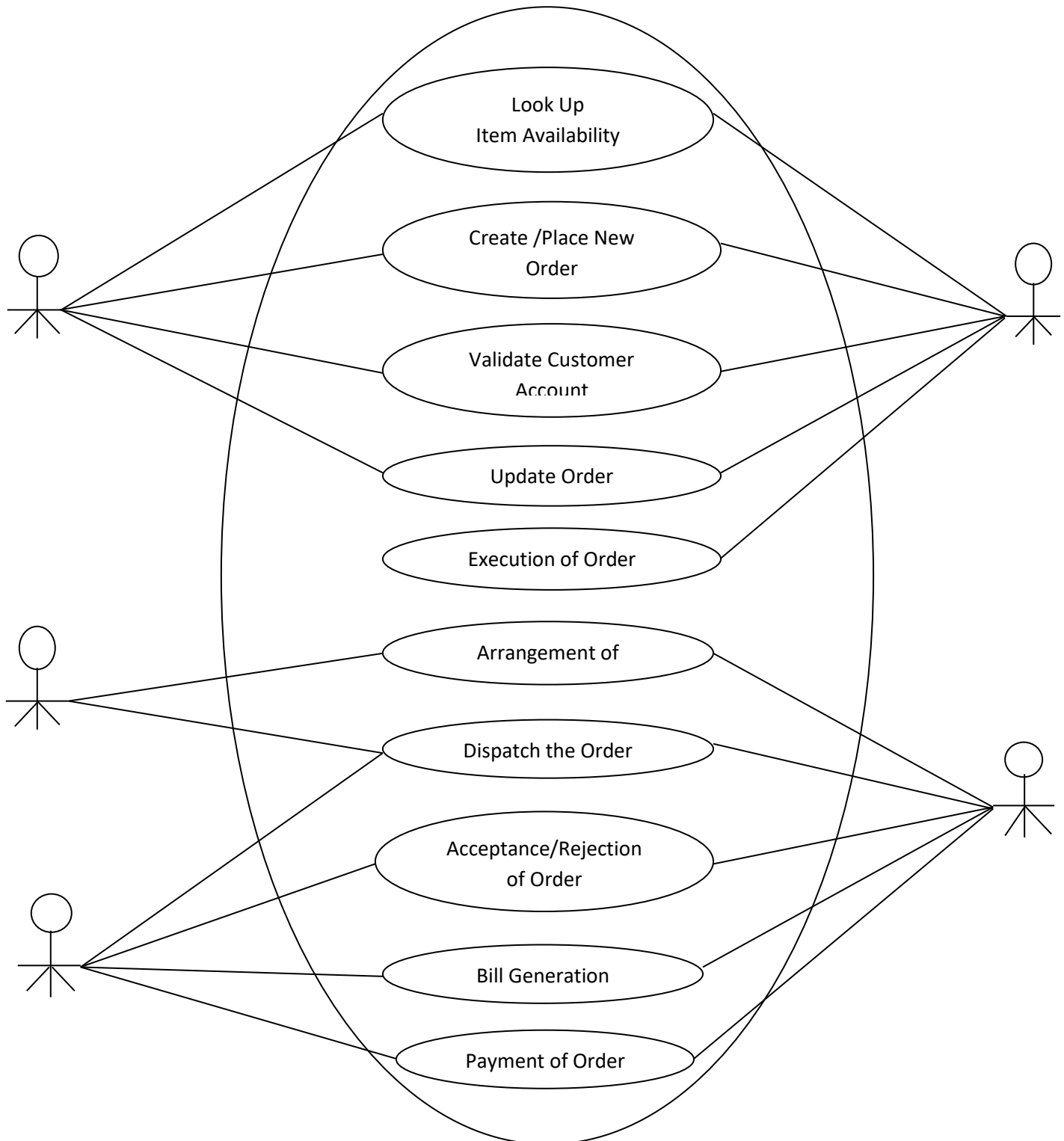
Report List

Sr. No.	Report Name	Purpose
1	MRN.java	This report shows purchase stock entry.
2	Order.java	This report show store the order information.
3	Payment.java	This report shows the order information.
4	Bill.java	This report shows the details of order bill.

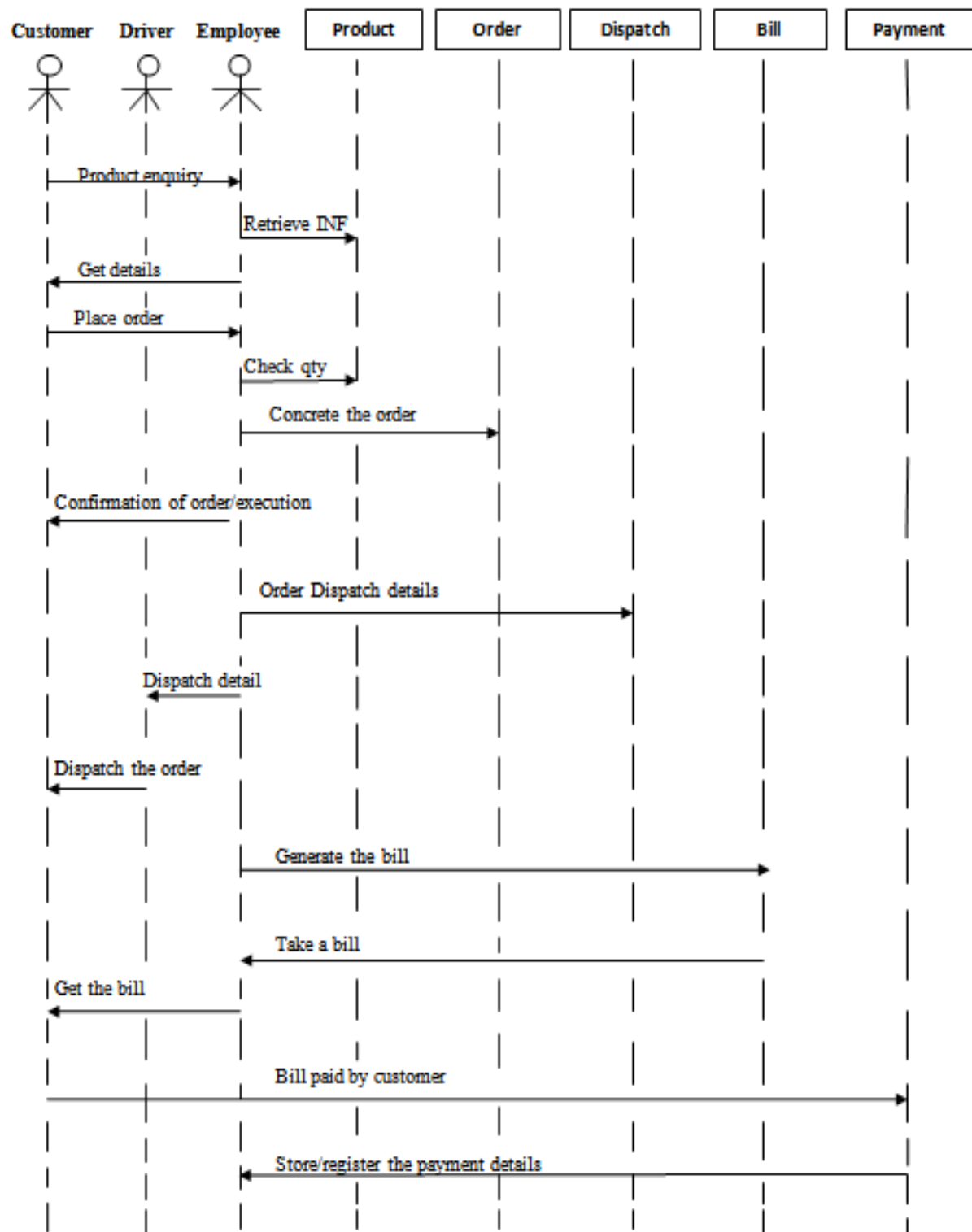
Chapter-5

SYSTEM DESIGN

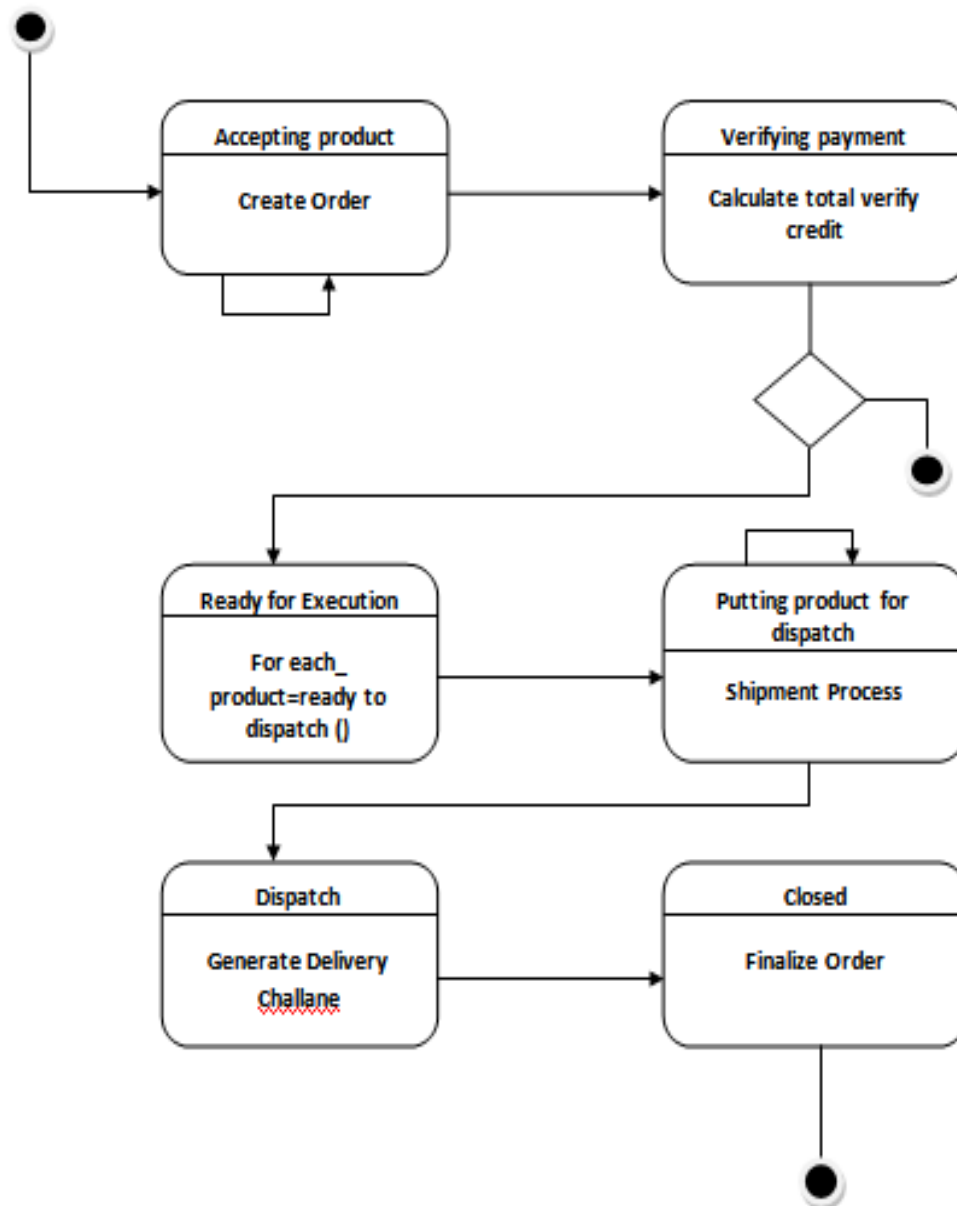
5.1 USE CASE DIAGRAM



5.2 SEQUENCE DIAGRAM



5.3 STATE DIAGRAM



5.4 TABLES

DATA DIRECTORY TABLE

Table Name: Login

Field Name	Data Type	Description
Username	VARCHAR	It stores the Username
Password	VARCHAR	It stores the Password

Table Name: Supplier master

Field Name	Data Type	Description
Supplier INTEGER	VARCHAR	It stores the unique Supplier INTEGER.
Name	VARCHAR	It stores the unique Supplier Name.
Address	VARCHAR	It stores the unique Supplier Address.
Phone INTEGER	INTEGER	It stores the contact INTEGER of supplier.
Account INTEGER	VARCHAR	It stores the supplier bank account INTEGER.
Credit	INTEGER	It stores the supplier credit details.
Debit	INTEGER	It stores the supplier debit details.
Balance	INTEGER	It stores the supplier balance details.

Table Name: Employee master

Field Name	Data Type	Description
Employee ID	VARCHAR	It stores the unique employee INTEGER.
Name	VARCHAR	It stores the unique employee Name.
Address	VARCHAR	It stores the unique employee Address.
Phone INTEGER	INTEGER	It stores the contact INTEGER of employee
Date of Birth	DATETIME	It stores the date of birth of an employee.
Date of Join	DATETIME	It stores the date of joining of an employee.
Email ID	VARCHAR	It stores the Email-id of employee.
Salary	INTEGER	It stores the salary of employee.

Table Name: Product master

Field Name	Data Type	Description
Product Code	VARCHAR	It stores the unique product code.
Name	VARCHAR	It stores the unique product name.
Description	VARCHAR	It stores the product description.
Cost Price	INTEGER	It stores the product cost price.

Sell Price	INTEGER	It stores the product selling price.
Discount	INTEGER	It stores the product limited discount.
Quantity	INTEGER	It stores the product quantity.
Reorder Level	INTEGER	It stores the reorder level.

Table Name: Customer master

Field Name	Data Type	Description
Customer INTEGER	VARCHAR	It stores the unique customer INTEGER.
Name	VARCHAR	It stores the unique customer Name.
Address	VARCHAR	It stores the unique customer Address.
Phone INTEGER	INTEGER	It stores the contact INTEGER of customer.
Account INTEGER	VARCHAR	It stores customer bank account INTEGER.
Credit	INTEGER	It stores the customer credit details.
Debit	INTEGER	It stores the customer debit details.
Balance	INTEGER	It stores the customer balance details.

Table Name: MRN master

Field Name	Data Type	Description
MRN INTEGER	VARCHAR	It stores the unique MRN INTEGER.
Date	DATETIME	It stores the date.
Supplier INTEGER	VARCHAR	It retrieve the supplier INTEGER.
Product Excise	VARCHAR	It stores product excise code.
Vehicle INTEGER	VARCHAR	It stores the vehicle INTEGER.
Driver Name	VARCHAR	It stores the driver name.

Table Name: MRN Transaction

Field Name	Data Type	Description
MRN INTEGER	VARCHAR	It stores the unique MRN INTEGER.
Product Code	VARCHAR	It retrieve the product code.
Quantity	INTEGER	It stores the product quantity.
Received Quantity	INTEGER	It stores the product received quantity.
Rejected Quantity	INTEGER	It stores the product rejected quantity.
Product Rate	INTEGER	It stores the single product rate.
Product Amount	INTEGER	It stores the total product amount.

Table Name: Order master

Field Name	Data Type	Description
------------	-----------	-------------

Order INTEGER	VARCHAR	It stores the unique order INTEGER.
Order Date	DATETIME	It stores the order date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Customer Name	VARCHAR	It retrieve the customer Name.

Table Name: Order Transaction

Field Name	Data Type	Description
Order INTEGER	VARCHAR	It retrieve order INTEGER.
Product Code	VARCHAR	It retrieve the product code.
Product Name	INTEGER	It stores the product name.
Product Description	INTEGER	It stores the product description.
Product Quantity	INTEGER	It stores the product quantity.
Product Rate	INTEGER	It stores the single product rate.
Order Billed	Yes/No	It stores record from order.

Table Name: Dispatch Master

Field Name	Data Type	Description
Dispatch INTEGER	VARCHAR	It stores the unique dispatch INTEGER.
Dispatch Date	DATETIME	It stores the dispatch date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Customer Address	VARCHAR	It retrieve the customer Name.
Order INTEGER	VARCHAR	It retrieve order INTEGER.
Vehicle INTEGER	VARCHAR	It stores the vehicle INTEGER.
Driver Name	VARCHAR	It stores the driver name.

Table Name: Dispatch Transaction

Field Name	Data Type	Description
Dispatch INTEGER	VARCHAR	It stores the unique dispatch INTEGER.
Product Code	VARCHAR	It retrieve the product code.
Product Quantity	INTEGER	It stores the product quantity.
Received Quantity	INTEGER	It stores the product received quantity.
Rejected Quantity	INTEGER	It stores the product rejected quantity.
Product Rate	INTEGER	It stores the product rate.

Table Name: Bill Master

Field Name	Data Type	Description
Bill INTEGER	VARCHAR	It stores the unique bill INTEGER.

Bill Date	DATETIME	It stores the bill generate date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Customer Name	VARCHAR	It stores the customer name.
Order INTEGER	VARCHAR	It retrieve the order number.
Dispatch INTEGER	VARCHAR	It retrieve the dispatch INTEGER.

Table Name: Bill Transaction

Field Name	Data Type	Description
Bill INTEGER	VARCHAR	It stores the unique bill INTEGER.
Product Code	VARCHAR	It retrieve the product code.
Product Description	INTEGER	It stores the product description.
Received Quantity	INTEGER	It stores the product received quantity.
Product Rate	INTEGER	It stores the product rate.
Product Discount	INTEGER	It stores the product limited discount.
Product Amount	INTEGER	It stores the total product amount.

Table Name: Product Amount

Field Name	Data Type	Description
Bill INTEGER	VARCHAR	It stores the unique bill INTEGER.
Bill Date	DATETIME	It stores the bill generate date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Order INTEGER	VARCHAR	It retrieve the order number.
Dispatch INTEGER	VARCHAR	It retrieve the dispatch INTEGER.
Product Amount	INTEGER	It stores the total product amount.

Table Name: Payment Receipt

Field Name	Data Type	Description
Receipt INTEGER	VARCHAR	It stores the unique receipt INTEGER.
Payment Date	DATETIME	It stores the payment generate date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Customer Name	VARCHAR	It stores the customer name.
Payment Against	VARCHAR	It check the payment against.
Bill Amount	INTEGER	It retrieve the payment bill amount.
Payment Mode	VARCHAR	It's type of payment mode.
Cheque INTEGER	INTEGER	It stores the cheque INTEGER.
Bank Name	VARCHAR	It stores the bank name.
Cheque Amount	INTEGER	It stores the cheque amount.

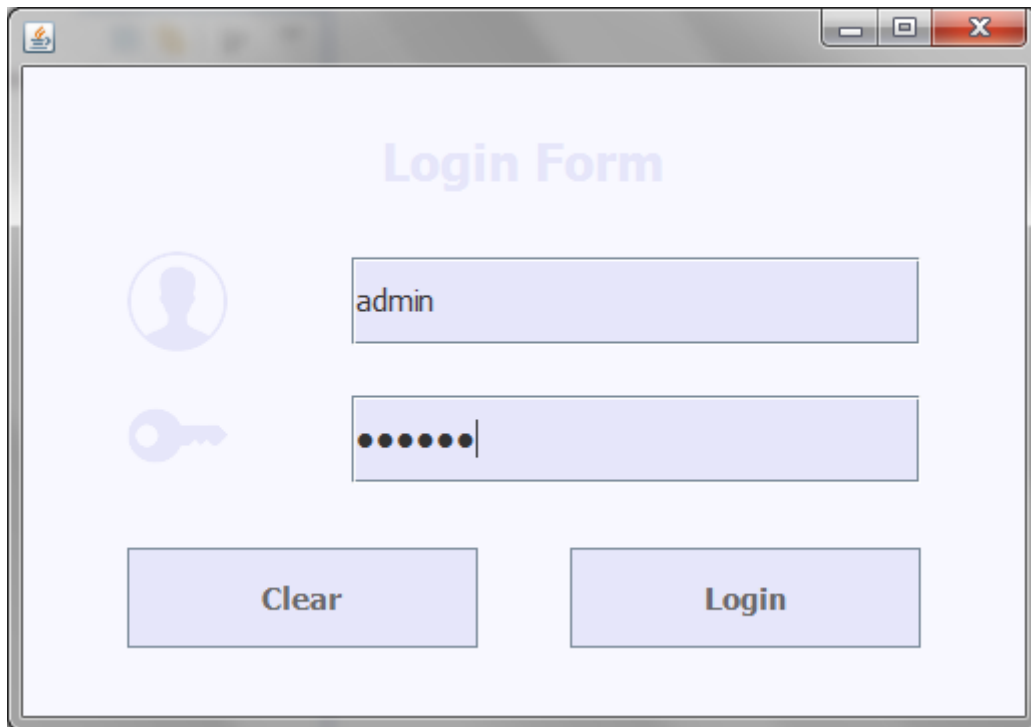
CHAPTER -6

SYSTEM CODINE & FORM LAYOUTS

Program Description With Naming Convention

- **Program Name:-** Login.java

Program Description:- This form is used for security with the help of password.



Code:-

```
package com.asms.form;
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import com.asms.form.db.DBLoginWindow;
import java.awt.event.*;
```

```
@SuppressWarnings("serial")
public class LoginWindow extends JFrame
{
    private JPanel contentPane;
    private JTextField txtLoginname;
    private JPasswordField txtPassword;
    private String name, pass;
```

```
public LoginWindow()
{
    setBounds(100, 100, 516, 362);
    //setAlwaysOnTop(true);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(0, 128, 0));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JPanel panel = new JPanel();
    panel.setLayout(null);
    panel.setBackground(null);
    panel.setBounds(10, 11, 480, 306);
    getContentPane().add(panel);

    txtLoginname = new JTextField();
    txtLoginname.setFont(new Font("Tahoma", Font.PLAIN, 15));
    txtLoginname.setColumns(10);
    txtLoginname.setBackground(new Color(230, 230, 250));
    txtLoginname.setBounds(196, 86, 227, 34);
    panel.add(txtLoginname);

    JButton btnLogin = new JButton("Login");
    btnLogin.addActionListener(new ActionListener()
    {
        @SuppressWarnings({ "deprecation" })
        public void actionPerformed(ActionEvent arg0)
        {
            name=txtLoginname.getText();
            pass=txtPassword.getText();
            DBLoginWindow db=new DBLoginWindow();
            int i = db.checkLogin(name,pass);
            if (i==0)
                setVisible(false);
        }
    });
    btnLogin.setForeground(SystemColor.controlDkShadow);
    btnLogin.setFont(new Font("Tahoma", Font.BOLD, 15));
    btnLogin.setBackground(new Color(230, 230, 250));
    btnLogin.setBounds(263, 229, 175, 50);
    panel.add(btnLogin);
}
```

```
txtPassword = new JPasswordField();
txtPassword.setFont(new Font("Tahoma", Font.BOLD, 15));
txtPassword.setBackground(new Color(230, 230, 250));
txtPassword.setBounds(196, 148, 227, 34);
panel.add(txtPassword);
```

```
JLabel label = new JLabel("Login Form");
label.setForeground(Color.DARK_GRAY);
label.setFont(new Font("Tahoma", Font.BOLD, 25));
label.setBackground(Color.BLACK);
label.setBounds(169, 11, 141, 50);
panel.add(label);
```

```
JButton btnClear = new JButton("Clear");
btnClear.setForeground(SystemColor.controlDkShadow);
btnClear.setFont(new Font("Tahoma", Font.BOLD, 15));
btnClear.setBackground(new Color(230, 230, 250));
btnClear.setBounds(42, 229, 175, 50);
panel.add(btnClear);
```

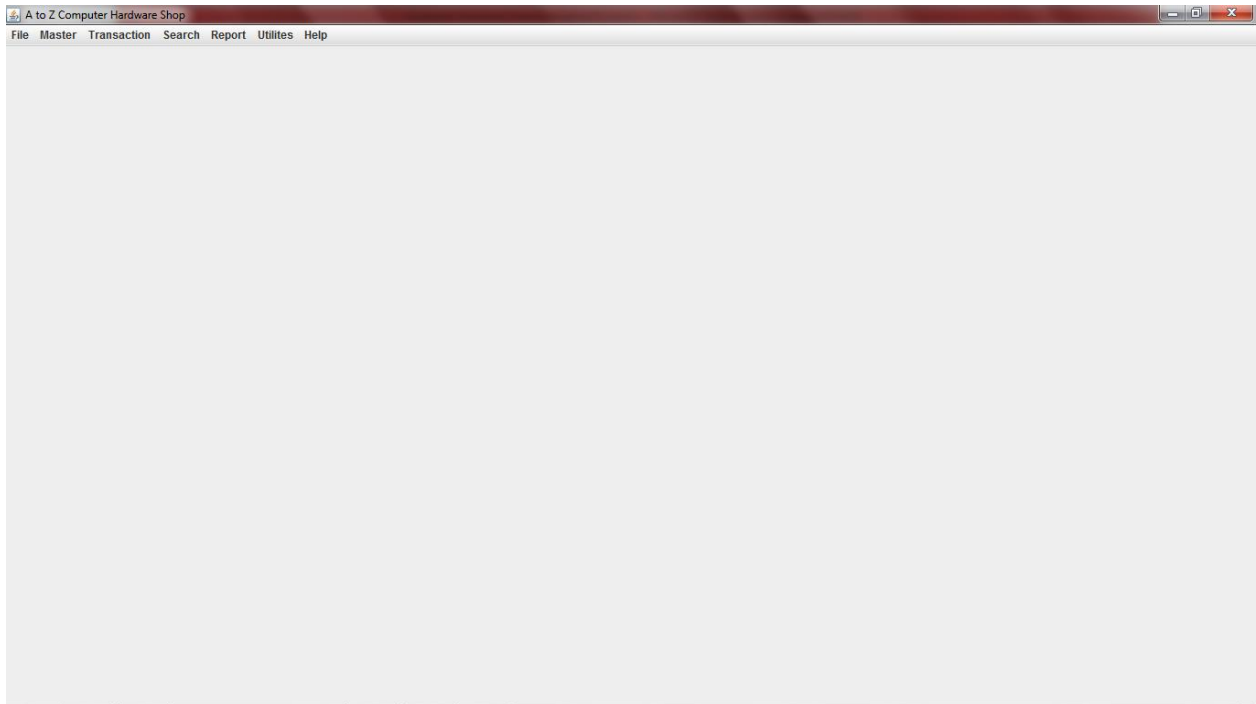
```
JLabel lblUsername = new JLabel("User Name :");
lblUsername.setForeground(Color.DARK_GRAY);
lblUsername.setFont(new Font("Times New Roman", Font.BOLD, 22));
lblUsername.setBackground(Color.BLACK);
lblUsername.setBounds(45, 78, 141, 50);
panel.add(lblUsername);
```

```
JLabel lblPassword = new JLabel("Password :");
lblPassword.setForeground(Color.DARK_GRAY);
lblPassword.setFont(new Font("Times New Roman", Font.BOLD, 22));
lblPassword.setBackground(Color.BLACK);
lblPassword.setBounds(42, 140, 141, 50);
panel.add(lblPassword);
```

```
btnClear.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        txtLoginname.setText(null);
        txtPassword.setText(null);
    }
});
}
```

- **Program Name:- Menuscreen.java**

Program Description:- This is the multiple documentation interfaces for the all of the form in the project.



Code:-

```
package com.chsm.form;
import java.awt.event.*;
import javax.swing.*;
import com.chsm.form.modify.*;
import com.chsm.form.report.*;
import com.chsm.form.transaction.*;

@SuppressWarnings("serial")
public class MenuScreenForm extends JFrame
{
    private JMenuBar menuBar;
    private JMenu mnFile, mnMaster, mnNewMenu, mnSearch, mnReport, mnUtilites,
mnHelp;
    private JMenuItem mntmPasswordChange, mntmLogoutUser, mntmSystemExit,
mntmAddEmployee, mntmAddSupplier, mntmAddProduct, mntmAddCustomer,
mntmSupplierOrder,
        mntmCustomerOrder, mntmDispatchProduct, mntmSearchSupplier,
mntmSearchCustomer, mntmSearchCustomerOrder, mntmSearchEmployee,
        mntmSearchSupplierOrder, mntmEmployeeReport,
mntmCustomerReport, mntmSupplierReport, mntmProductReport,
mntmCustOrderReport, mntmNotepad,
        mntmCalculator, mntmSystemInfo, mntmContactUs;
    private JSeparator s1, s2;
```

```
private JMenuItem mntmEmployeepayment;
private JMenuItem mntmEmployeepayment
public MenuScreenForm()
{
    setTitle("A to Z Computer Hardware Shop");
    setBounds(100, 100, 816, 632);
    setExtendedState(JFrame.MAXIMIZED_BOTH);

    menuBar = new JMenuBar();
    setJMenuBar(menuBar);

    mnFile = new JMenu("File");
    menuBar.add(mnFile);

    mntmPasswordChange = new JMenuItem("Password Change");
    mntmPasswordChange.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
        {
            new PasswordChange().setVisible(true);
        }
    });
    mnFile.add(mntmPasswordChange);

    mntmLogoutUser = new JMenuItem("Logout User");
    mntmLogoutUser.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
        {
            setVisible(false);
            new LoginWindow().setVisible(true);
        }
    });
    mnFile.add(mntmLogoutUser);

    s1 = new JSeparator();
    mnFile.add(s1);

    mntmSystemExit = new JMenuItem("System Exit");
    mntmSystemExit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
        }
    });
    mnFile.add(mntmSystemExit);

    mnMaster = new JMenu("Master");
    menuBar.add(mnMaster);
```



```
mntmAddEmployee = new JMenuItem("Add_Employee");
mntmAddEmployee.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new AddEmployee().setVisible(true);
    }
});
mnMaster.add(mntmAddEmployee);

mntmAddSupplier = new JMenuItem("Add_Supplier");
mntmAddSupplier.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new AddSupplier().setVisible(true);
    }
});
mnMaster.add(mntmAddSupplier);

mntmAddProduct = new JMenuItem("Add_Product");
mntmAddProduct.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new AddProduct().setVisible(true);
    }
});
mnMaster.add(mntmAddProduct);

mntmAddCustomer = new JMenuItem("Add_Customer");
mntmAddCustomer.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new AddCustomer().setVisible(true);
    }
});
mnMaster.add(mntmAddCustomer);

mnNewMenu = new JMenu("Transaction");
menuBar.add(mnNewMenu);

mntmSupplierOrder = new JMenuItem("Supplier_Order");
mntmSupplierOrder.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new SupplierOrder().setVisible(true);
    }
});
```

```
    }  
});  
mnNewMenu.add(mntmSupplierOrder);  
mntmCustomerOrder = new JMenuItem("Customer_Order");  
mntmCustomerOrder.addActionListener(new ActionListener()  
{  
    public void actionPerformed(ActionEvent arg0)  
    {  
        new CustomerOrder().setVisible(true);  
    }  
});  
mnNewMenu.add(mntmCustomerOrder);  
  
mntmDispatchProduct = new JMenuItem("Dispatch_Product");  
mntmDispatchProduct.addActionListener(new ActionListener()  
{  
    public void actionPerformed(ActionEvent arg0)  
    {  
        new DispatchCustomerOrder().setVisible(true);  
    }  
});  
mnNewMenu.add(mntmDispatchProduct);  
  
mntmEmployeepayment = new JMenuItem("Employee_Payment");  
mntmEmployeepayment.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        new EmployeePayment().setVisible(true);  
    }  
});  
mnNewMenu.add(mntmEmployeepayment);  
mnSearch = new JMenu("Search");  
menuBar.add(mnSearch);  
  
mntmSearchEmployee = new JMenuItem("Search_Employee");  
mntmSearchEmployee.addActionListener(new ActionListener()  
{  
    public void actionPerformed(ActionEvent arg0)  
    {  
        new ModifyEmployee().setVisible(true);  
    }  
});  
mnSearch.add(mntmSearchEmployee);  
  
mntmSearchSupplier = new JMenuItem("Search_Supplier");  
mntmSearchSupplier.addActionListener(new ActionListener()  
{  
    public void actionPerformed(ActionEvent arg0)  
    {  
        new ModifySupplier().setVisible(true);
```

```
    }
});
mnSearch.add(mntmSearchSupplier);

mntmSearchCustomer = new JMenuItem("Search_Customer");
mntmSearchCustomer.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new ModifyCustomer().setVisible(true);
    }
});
mnSearch.add(mntmSearchCustomer);

mntmSearchCustomerOrder = new JMenuItem("Search_Customer_Order");
mntmSearchCustomerOrder.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new ModifyCustomerOrder().setVisible(true);
    }
});

mntmSearchSupplierOrder = new JMenuItem("Search_Supplier_Order");
mntmSearchSupplierOrder.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new ModifySupplierOrder().setVisible(true);
    }
});

s2 = new JSeparator();
mnSearch.add(s2);
mnSearch.add(mntmSearchSupplierOrder);
mnSearch.add(mntmSearchCustomerOrder);

mnReport = new JMenu("Report");
menuBar.add(mnReport);

mntmEmployeeReport = new JMenuItem("Employee_Report");
mntmEmployeeReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdEmployeeInfo().setVisible(true);
    }
});
mnReport.add(mntmEmployeeReport);
```

```
mntmCustomerReport = new JMenuItem("Customer_Report");
mntmCustomerReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdCustomerInfo().setVisible(true);;
    }
});
mnReport.add(mntmCustomerReport);

mntmSupplierReport = new JMenuItem("Supplier_Report");
mntmSupplierReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdSupplierInfo().setVisible(true);;
    }
});
mnReport.add(mntmSupplierReport);

mntmProductReport = new JMenuItem("Product_Report");
mntmProductReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdProductInfo().setVisible(true);;
    }
});
mnReport.add(mntmProductReport);

mntmCustOrderReport = new JMenuItem("Customer_Order_Report");
mntmCustOrderReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdCustomerOrder().setVisible(true);
    }
});

JSeparator separator = new JSeparator();
mnReport.add(separator);
mnReport.add(mntmCustOrderReport);

JMenuItem mntmSupplierOredrReport = new
JMenuItem("Supplier_Oredr_Report");
mntmSupplierOredrReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdSupplierOrder().setVisible(true);
    }
});
mnReport.add(mntmSupplierOredrReport);

mntmEmployeePaymnet = new JMenuItem("Employee_Paymnet");
mntmEmployeePaymnet.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdEmpPayment().setVisible(true);
    }
})
```

```
});
mntmReport.add(mntmEmployeePaymnet);

mnUtilites = new JMenu("Utilites");
menuBar.add(mnUtilites);

mntmNotepad = new JMenuItem("Notepad");
mntmNotepad.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        try
        {
            Runtime.getRuntime().exec("Notepad.exe");
        }
        catch(Exception ex1)
        {
            JOptionPane.showMessageDialog(null, "Error, Cannot
start notepad", "Application Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});
mnUtilites.add(mntmNotepad);

mntmCalculator = new JMenuItem("Calculator");
mntmCalculator.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        try
        {
            Runtime.getRuntime().exec("calc.exe");
        }
        catch(Exception ex1)
        {
            JOptionPane.showMessageDialog(null, "Error, Cannot
start calculator", "Application Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});
mnUtilites.add(mntmCalculator);

mnHelp = new JMenu("Help");
menuBar.add(mnHelp);

mntmSystemInfo = new JMenuItem("System_Info");
mnHelp.add(mntmSystemInfo);

mntmContactUs = new JMenuItem("Contact_Us");
```

```

        mnHelp.add(mntmContactUs);
        getContentPane().setLayout(null);

    }
}

```

- **Program Name:-** AddEmployee.java

Program Description:- This form is used to Add Employee information.

Code:-

```

package com.chsm.form;
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.sql.*;

import com.chsm.form.db.DBEmployee;
import com.toedter.calendar.JDateChooser;

@SuppressWarnings("serial")
public class AddEmployee extends JFrame {

    private JPanel panel, panelEmpppi, panelEmpoc, panelOperation;
    private JTextField txtEmployeeID, txtEmployeeName, txtEmployeeContact,
txtEmployeeEmailID, txtEmployeeDesignation, txtEmployeeSalary, txtEmployeeUserName,
txtEmployeePassword;
    private JLabel lblEmployeeId, lblEmployeeName, lblEmployeeBirthDate,
lblEmployeeGender, lblEmployeeAddress, lblEmployeeContact, lblEmployeeEmailID,
lblEmployeePassword;

```

```
private JLabel lblEmployeeJoiningDate, lblEmployeeDesignation,
lblEmployeeSalary, lblEmployeeUserName;
private JRadioButton rbMale, rbFemale;
private JButton btnSave, btnClear, btnExit;
private ButtonGroup buttonGroup;
private JTextArea txtaEmployeeAddress;
private JScrollPane scrollPane;
private JDateChooser txtEmployeeBirthDate, txtEmployeeJoiningDate;
private DBEmployee db = new DBEmployee();

public static java.sql.Date datemethad(java.util.Date date)
{
    if(date != null)
    {
        java.sql.Date sqld = new Date(date.getTime());
        return sqld;
    }
    return null;
}

public AddEmployee()
{
    setResizable(false);
    //setAlwaysOnTop(true);
    setTitle("Adding_Employee_Information");
    setBounds(100, 100, 1200, 525);
    panel = new JPanel();
    panel.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(panel);
    panel.setLayout(null);

    panelEmppi = new JPanel();
    panelEmppi.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Employee_Personal_Information", TitledBorder.LEADING, TitledBorder.TOP, null,
null));
    panelEmppi.setBounds(22, 19, 564, 458);
    panel.add(panelEmppi);
    panelEmppi.setLayout(null);

    lblEmployeeId = new JLabel("Employee ID :");
    lblEmployeeId.setFont(new Font("Tahoma", Font.PLAIN, 15));
    lblEmployeeId.setBounds(38, 44, 150, 25);
```

```
panelEmpppi.add(lblEmployeeId);

txtEmployeeID = new JTextField();
txtEmployeeID.setColumns(10);
txtEmployeeID.setBounds(253, 44, 270, 25);
panelEmpppi.add(txtEmployeeID);
txtEmployeeID.setText(db.setID());
txtEmployeeID.setEditable(false);
lblEmployeeName = new JLabel("Employee Name :");
lblEmployeeName.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeName.setBounds(38, 89, 150, 25);
panelEmpppi.add(lblEmployeeName);

txtEmployeeName = new JTextField();
txtEmployeeName.setColumns(10);
txtEmployeeName.setBounds(253, 89, 270, 25);
panelEmpppi.add(txtEmployeeName);

lblEmployeeBirthDate = new JLabel("Employee Birth Date :");
lblEmployeeBirthDate.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeBirthDate.setBounds(38, 134, 150, 25);
panelEmpppi.add(lblEmployeeBirthDate);

lblEmployeeGender = new JLabel("Employee Gender :");
lblEmployeeGender.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeGender.setBounds(38, 183, 150, 25);
panelEmpppi.add(lblEmployeeGender);

buttonGroup = new ButtonGroup();
rbMale = new JRadioButton("Male");
rbMale.setBackground(null);
buttonGroup.add(rbMale);
rbMale.setHorizontalAlignment(SwingConstants.CENTER);
rbMale.setFont(new Font("Tahoma", Font.PLAIN, 15));
rbMale.setBounds(273, 184, 64, 23);
panelEmpppi.add(rbMale);
rbMale.setActionCommand("Male");

rbFemale = new JRadioButton("Female");
rbFemale.setBackground(null);
buttonGroup.add(rbFemale);
rbFemale.setHorizontalAlignment(SwingConstants.CENTER);
rbFemale.setFont(new Font("Tahoma", Font.PLAIN, 15));
```



```
rbFemale.setBounds(388, 184, 73, 23);  
panelEmpppi.add(rbFemale);  
rbFemale.setActionCommand("Female");
```

```
lblEmployeeAddress = new JLabel("Employee Address :");  
lblEmployeeAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblEmployeeAddress.setBounds(38, 232, 150, 25);  
panelEmpppi.add(lblEmployeeAddress);
```

```
lblEmployeeContact = new JLabel("Employee Contact :");  
lblEmployeeContact.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblEmployeeContact.setBounds(38, 345, 150, 25);  
panelEmpppi.add(lblEmployeeContact);
```

```
txtEmployeeContact = new JTextField();  
txtEmployeeContact.setColumns(10);  
txtEmployeeContact.setBounds(253, 345, 270, 25);  
panelEmpppi.add(txtEmployeeContact);
```

```
lblEmployeeEmailID = new JLabel("Employee Email ID :");  
lblEmployeeEmailID.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblEmployeeEmailID.setBounds(38, 390, 150, 25);  
panelEmpppi.add(lblEmployeeEmailID);
```

```
txtEmployeeEmailID = new JTextField();  
txtEmployeeEmailID.setColumns(10);  
txtEmployeeEmailID.setBounds(253, 390, 270, 25);  
panelEmpppi.add(txtEmployeeEmailID);
```

```
scrollPane = new JScrollPane();  
scrollPane.setBounds(253, 234, 270, 90);  
panelEmpppi.add(scrollPane);
```

```
txtaEmployeeAddress = new JTextArea();  
scrollPane.setViewportView(txtaEmployeeAddress);
```

```
txtEmployeeBirthDate = new JDateChooser();  
txtEmployeeBirthDate.setBounds(253, 134, 270, 25);  
panelEmpppi.add(txtEmployeeBirthDate);
```

```
panelEmpoc = new JPanel();
```

```
panelEmpoc.setBorder(new  
TitledBorder(UIManager.getBorder("TitledBorder.border"),  
"Employee_Official_Information", TitledBorder.LEADING, TitledBorder.TOP, null, null));  
panelEmpoc.setBounds(608, 19, 564, 289);  
panel.add(panelEmpoc);  
panelEmpoc.setLayout(null);
```

```
lblEmployeeJoiningDate = new JLabel("Employee Joining Date :");  
lblEmployeeJoiningDate.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblEmployeeJoiningDate.setBounds(38, 27, 169, 25);  
panelEmpoc.add(lblEmployeeJoiningDate);
```

```
lblEmployeeDesignation = new JLabel("Employee Designation :");  
lblEmployeeDesignation.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblEmployeeDesignation.setBounds(38, 79, 169, 25);  
panelEmpoc.add(lblEmployeeDesignation);
```

```
txtEmployeeDesignation = new JTextField();  
txtEmployeeDesignation.setColumns(10);  
txtEmployeeDesignation.setBounds(253, 79, 270, 25);  
panelEmpoc.add(txtEmployeeDesignation);
```

```
lblEmployeeSalary = new JLabel("Employee Salary :");  
lblEmployeeSalary.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblEmployeeSalary.setBounds(38, 235, 150, 25);  
panelEmpoc.add(lblEmployeeSalary);
```

```
txtEmployeeSalary = new JTextField();  
txtEmployeeSalary.setColumns(10);  
txtEmployeeSalary.setBounds(253, 235, 270, 25);  
panelEmpoc.add(txtEmployeeSalary);
```

```
lblEmployeeUserName = new JLabel("Employee User Name :");  
lblEmployeeUserName.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblEmployeeUserName.setBounds(38, 131, 169, 25);  
panelEmpoc.add(lblEmployeeUserName);
```

```
txtEmployeeUserName = new JTextField();  
txtEmployeeUserName.setColumns(10);  
txtEmployeeUserName.setBounds(253, 131, 270, 25);  
panelEmpoc.add(txtEmployeeUserName);
```

```
lblEmployeePassword = new JLabel("Employee Password :");
```

```
lblEmployeePassword.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeePassword.setBounds(38, 183, 169, 25);
panelEmpoc.add(lblEmployeePassword);

txtEmployeePassword = new JTextField();
txtEmployeePassword.setColumns(10);
txtEmployeePassword.setBounds(253, 183, 270, 25);
panelEmpoc.add(txtEmployeePassword);

txtEmployeeJoiningDate = new JDateChooser();
txtEmployeeJoiningDate.setBackground(SystemColor.text);
txtEmployeeJoiningDate.setBounds(253, 27, 270, 25);
panelEmpoc.add(txtEmployeeJoiningDate);
panelOperation = new JPanel();
panelOperation.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Operation ",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelOperation.setBounds(608, 327, 564, 150);
panel.add(panelOperation);
panelOperation.setLayout(null);

btnSave = new JButton("Save");
btnSave.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        db.addemployee(Integer.parseInt(txtEmployeeID.getText()),
txtEmployeeContact.getText(), Integer.parseInt(txtEmployeeSalary.getText()),
txtEmployeeName.getText(),
buttonGroup.getSelection().getActionCommand(), txtaEmployeeAddress.getText(),
txtEmployeeEmailID.getText(),
txtEmployeeDesignation.getText(),
datemethad(txtEmployeeBirthDate.getDate()),
datemethad(txtEmployeeJoiningDate.getDate()));

        db.addlogin(Integer.parseInt(txtEmployeeID.getText()),
txtEmployeeUserName.getText(), txtEmployeePassword.getText());
        JOptionPane.showMessageDialog(null, "Employee Data Save
Sucesfull....");
        dispose();
    }
});
btnSave.setFont(new Font("Tahoma", Font.PLAIN, 15));
```

```
btnSave.setBackground(new Color(230, 230, 250));
btnSave.setBounds(28, 50, 150, 50);
panelOperation.add(btnSave);

btnClear = new JButton("Clear");
btnClear.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        txtEmployeeName.setText(null);
        txtEmployeeContact.setText(null);
        txtEmployeeEmailID.setText(null);
        txtEmployeeDesignation.setText(null);
        txtEmployeeSalary.setText(null);
        txtEmployeeUserName.setText(null);
        txtEmployeePassword.setText(null);
        txtaEmployeeAddress.setText(null);
        buttonGroup.clearSelection();
        txtEmployeeBirthDate.setCalendar(null);
        txtEmployeeJoiningDate.setCalendar(null);
    }
});
btnClear.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnClear.setBackground(new Color(230, 230, 250));
btnClear.setBounds(206, 50, 150, 50);
panelOperation.add(btnClear);

btnExit = new JButton("Exit");
btnExit.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        JOptionPane.showMessageDialog(null, "Do you want to close
Add_Employee....");
        dispose();
    }
});
btnExit.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnExit.setBackground(new Color(230, 230, 250));
btnExit.setBounds(384, 50, 150, 50);
panelOperation.add(btnExit);
```

```

    }
}

```

- **Program Name:-** ModifyCustomer.java

Program Description:- This form is used to update the customer's information.

Code:-

```

package com.chsm.form.modify;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.border.*;
import com.chsm.form.db.DBCustomer;
import com.toedter.calendar.JDateChooser;
import DBmodel.sqlconnection;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

```

```

@SuppressWarnings("serial")
public class ModifyCustomer extends JFrame
{
    private JPanel panelCustoi, panel, panelCustpi, panelOperation, panelCustomerS;

```

```
private JLabel lblCustomerId, lblCustomerEmailId, lblCustomerCreditAmount,
lblCustomerContact, lblCustomerDebitAmount, lblCustomerName,
lblCustomerBalanceAmount, lblCustomerRegisteredDate, lblCustomerGender,
lblCustomerAddress;
private JTextField txtCustomerName, txtCustomerBAmount, txtCustomerDeAmount,
txtCustomerCrAmount, txtCustomerEmailId, txtCustomerContact;
private JDateChooser txtCustomerRDate;
@SuppressWarnings("rawtypes")
private JComboBox txtCustomerId;
private ButtonGroup buttonGroup;
private JRadioButton rbMale, rbFemale;
private JScrollPane scrollPane;
private JTextArea txtCustomerAddress;
private JButton btnExit, btnDelete, btnUpadate, btnClear;
private int cid;
private DBCustomer db = new DBCustomer();

public static java.sql.Date datemethad(java.util.Date date)
{
    if(date != null)
    {
        java.sql.Date sqld = new Date(date.getTime());
        return sqld;
    }
    return null;
}

@SuppressWarnings("unchecked")
public void data()
{
    try
    {
        Connection c =sqlconnection.getConnection();
        String sql = "Select * from customer";
        PreparedStatement pst = c.prepareStatement(sql);
        ResultSet rs = pst.executeQuery();
        if(!rs.isBeforeFirst())
        {
            JOptionPane.showMessageDialog(null, "there is no data of
Customer in table....");
        }
    }
}
```

```
        else
        {
            while(rs.next())
            {
                txtCustomerId.addItem(rs.getInt("id"));
            }
        }
        pst.close();
    } catch (SQLException e) { e.printStackTrace(); }
}

@SuppressWarnings("rawtypes")
public ModifyCustomer()
{

    setResizable(false);
    setTitle("Searching_Customer_Information");
    setBounds(100, 100, 1200, 525);
    panel = new JPanel();
    panel.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(panel);
    panel.setLayout(null);

    panelCustpi = new JPanel();
    panelCustpi.setLayout(null);
    panelCustpi.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Customer_Personal_Information", TitledBorder.LEADING, TitledBorder.TOP, null, null));
    panelCustpi.setBounds(22, 163, 564, 307);
    panel.add(panelCustpi);

    lblCustomerRegisteredDate = new JLabel("Customer Registered Date :");
    lblCustomerRegisteredDate.setForeground(Color.BLACK);
    lblCustomerRegisteredDate.setFont(new Font("Tahoma", Font.PLAIN, 15));
    lblCustomerRegisteredDate.setBounds(35, 24, 189, 25);
    panelCustpi.add(lblCustomerRegisteredDate);

    lblCustomerGender = new JLabel("Customer Gender :");
    lblCustomerGender.setForeground(Color.BLACK);
    lblCustomerGender.setFont(new Font("Tahoma", Font.PLAIN, 15));
    lblCustomerGender.setBounds(35, 68, 189, 25);
    panelCustpi.add(lblCustomerGender);

    buttonGroup = new ButtonGroup();
```

```
rbMale = new JRadioButton("Male");
rbMale.setBackground(null);
buttonGroup.add(rbMale);
rbMale.setHorizontalAlignment(SwingConstants.CENTER);
rbMale.setFont(new Font("Tahoma", Font.PLAIN, 15));
rbMale.setBounds(273, 69, 64, 23);
panelCustpi.add(rbMale);
rbMale.setActionCommand("Male");

rbFemale = new JRadioButton("Female");
rbFemale.setBackground(null);
buttonGroup.add(rbFemale);
rbFemale.setHorizontalAlignment(SwingConstants.CENTER);
rbFemale.setFont(new Font("Tahoma", Font.PLAIN, 15));
rbFemale.setBounds(388, 69, 73, 23);
panelCustpi.add(rbFemale);
rbFemale.setActionCommand("Female");

lblCustomerAddress = new JLabel("Customer Address :");
lblCustomerAddress.setForeground(Color.BLACK);
lblCustomerAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerAddress.setBounds(35, 110, 189, 25);
panelCustpi.add(lblCustomerAddress);

lblCustomerContact = new JLabel("Customer Contact :");
lblCustomerContact.setForeground(Color.BLACK);
lblCustomerContact.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerContact.setBounds(35, 219, 189, 25);
panelCustpi.add(lblCustomerContact);

txtCustomerContact = new JTextField();
txtCustomerContact.setEditable(true);
txtCustomerContact.setColumns(10);
txtCustomerContact.setBackground(new Color(230, 230, 250));
txtCustomerContact.setBounds(259, 219, 270, 25);
panelCustpi.add(txtCustomerContact);

lblCustomerEmailId = new JLabel("Customer Email ID :");
lblCustomerEmailId.setForeground(Color.BLACK);
lblCustomerEmailId.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerEmailId.setBounds(35, 263, 189, 25);
panelCustpi.add(lblCustomerEmailId);
```



```
txtCustomerEmailId = new JTextField();
txtCustomerEmailId.setEditable(true);
txtCustomerEmailId.setColumns(10);
txtCustomerEmailId.setBackground(new Color(230, 230, 250));
txtCustomerEmailId.setBounds(259, 263, 270, 25);
panelCustpi.add(txtCustomerEmailId);
```

```
scrollPane = new JScrollPane();
scrollPane.setBounds(259, 110, 270, 90);
panelCustpi.add(scrollPane);
```

```
txtCustomerAddress = new JTextArea();
scrollPane.setViewportViewView(txtCustomerAddress);
```

```
txtCustomerRDate = new JDateChooser();
txtCustomerRDate.setBounds(259, 24, 262, 25);
panelCustpi.add(txtCustomerRDate);
```

```
panelCustoi = new JPanel();
panelCustoi.setLayout(null);
panelCustoi.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Customer_Official_Information", TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelCustoi.setBounds(608, 20, 564, 244);
panel.add(panelCustoi);
```

```
lblCustomerCreditAmount = new JLabel("Customer Credit Amount :");
lblCustomerCreditAmount.setForeground(Color.BLACK);
lblCustomerCreditAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerCreditAmount.setBounds(38, 42, 189, 25);
panelCustoi.add(lblCustomerCreditAmount);
```

```
txtCustomerCrAmount = new JTextField();
txtCustomerCrAmount.setEditable(true);
txtCustomerCrAmount.setColumns(10);
txtCustomerCrAmount.setBackground(new Color(230, 230, 250));
txtCustomerCrAmount.setBounds(253, 42, 270, 25);
panelCustoi.add(txtCustomerCrAmount);
```

```
lblCustomerDebitAmount = new JLabel("Customer Debit Amount :");
lblCustomerDebitAmount.setForeground(Color.BLACK);
lblCustomerDebitAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
```

```
lblCustomerDebitAmount.setBounds(38, 109, 189, 25);
panelCustoi.add(lblCustomerDebitAmount);

txtCustomerDeAmount = new JTextField();
txtCustomerDeAmount.setEditable(true);
txtCustomerDeAmount.setText("0");
txtCustomerDeAmount.setColumns(10);
txtCustomerDeAmount.setBackground(new Color(230, 230, 250));
txtCustomerDeAmount.setBounds(253, 109, 270, 25);
panelCustoi.add(txtCustomerDeAmount);
lblCustomerBalanceAmount = new JLabel("Customer Balance Amount :");
lblCustomerBalanceAmount.setForeground(Color.BLACK);
lblCustomerBalanceAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerBalanceAmount.setBounds(38, 176, 189, 25);
panelCustoi.add(lblCustomerBalanceAmount);

txtCustomerBAmount = new JTextField();
txtCustomerBAmount.setEditable(false);
txtCustomerBAmount.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        int i=(Integer.parseInt(txtCustomerCrAmount.getText())-
Integer.parseInt(txtCustomerDeAmount.getText()));
        txtCustomerBAmount.setText(String.valueOf(i));
    }
});
txtCustomerBAmount.setColumns(10);
txtCustomerBAmount.setBackground(new Color(230, 230, 250));
txtCustomerBAmount.setBounds(253, 176, 270, 25);
panelCustoi.add(txtCustomerBAmount);

panelOperation = new JPanel();
panelOperation.setLayout(null);
panelOperation.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Operation ",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelOperation.setBounds(608, 275, 564, 195);
panel.add(panelOperation);

btnUpdate = new JButton("Update ");
btnUpdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
```

```
        int i =
Integer.parseInt((txtCustomerId.getSelectedItem()).toString());
        db.updatecust(i, txtCustomerName.getText(),
datemethad(txtCustomerRDate.getDate()), buttonGroup.getSelection().getActionCommand(),
        txtCustomerAddress.getText(),
txtCustomerContact.getText(), txtCustomerEmailId.getText(),
Integer.parseInt(txtCustomerCrAmount.getText()),

Integer.parseInt(txtCustomerBAmount.getText()));
        JOptionPane.showMessageDialog(null, "Update Customer
Data Sucesfull....");
        dispose();
    }
});
btnUupdate.setForeground(Color.BLACK);
btnUupdate.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnUupdate.setBackground(new Color(230, 230, 250));
btnUupdate.setBounds(88, 31, 150, 50);
panelOperation.add(btnUupdate);

btnDelete = new JButton("Delete");
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int i =
Integer.parseInt((txtCustomerId.getSelectedItem()).toString());
        db.dlcust(i);
        JOptionPane.showMessageDialog(null, "Employee Data delete
Successfully....");
        dispose();
    }
});
btnDelete.setForeground(Color.BLACK);
btnDelete.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnDelete.setBackground(new Color(230, 230, 250));
btnDelete.setBounds(88, 112, 150, 50);
panelOperation.add(btnDelete);

btnClear = new JButton("Clear");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        txtCustomerName.setText(null);
        txtCustomerRDate.setDate(null);
        txtCustomerContact.setText(null);
```

```
        txtCustomerAddress.setText(null);
        txtCustomerContact.setText(null);
        txtCustomerEmailId.setText(null);
        txtCustomerCrAmount.setText(null);
        txtCustomerDeAmount.setText("0");
        txtCustomerBAmount.setText(null);
        buttonGroup.clearSelection();
    }
});
btnClear.setForeground(Color.BLACK);
btnClear.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnClear.setBackground(new Color(230, 230, 250));
btnClear.setBounds(326, 31, 150, 50);
panelOperation.add(btnClear);

btnExit = new JButton("Exit");
btnExit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        JOptionPane.showMessageDialog(null, "Do you want to close
Modify Customer....");
        dispose();
    }
});
btnExit.setForeground(Color.BLACK);
btnExit.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnExit.setBackground(new Color(230, 230, 250));
btnExit.setBounds(326, 112, 150, 50);
panelOperation.add(btnExit);

panelCustomerS = new JPanel();
panelCustomerS.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Customer_Search",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelCustomerS.setBounds(22, 20, 564, 132);
panel.add(panelCustomerS);
panelCustomerS.setLayout(null);

lblCustomerId = new JLabel("Customer ID :");
lblCustomerId.setForeground(Color.BLACK);
lblCustomerId.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerId.setBounds(35, 27, 189, 25);
panelCustomerS.add(lblCustomerId);
```

```
txtCustomerId = new JComboBox();
txtCustomerId.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

cid=Integer.parseInt(txtCustomerId.getSelectedItem().toString());

        try
        {
            final Connection c =sqlconnection.getConnection();
            final String sql = "Select * from customer where id=?";
            final PreparedStatement pst = c.prepareStatement(sql);
            pst.setInt(1, cid);
            final ResultSet rs = pst.executeQuery();
            while(rs.next())
            {

txtCustomerName.setText(rs.getString("name"));

txtCustomerRDate.setDate(rs.getDate("reg_date"));

txtCustomerContact.setText(rs.getString("contact"));

txtCustomerAddress.setText(rs.getString("address"));

txtCustomerContact.setText(rs.getString("contact"));

txtCustomerEmailId.setText(rs.getString("email"));

txtCustomerCrAmount.setText(rs.getString("balance"));
                final String gender1=rs.getString("gender");
                if(gender1.equals("Male"))
                {
                    rbMale.setSelected(true);
                }
                else
                {
                    rbFemale.setSelected(true);
                }
            }
        }
        catch(final Exception ex)
        {
            ex.printStackTrace();
        }
    }
}
```

```

        }
    }

    txtCustomerId.setBounds(259, 27, 270, 25);
    panelCustomerS.add(txtCustomerId);

    lblCustomerName = new JLabel("Customer Name :");
    lblCustomerName.setForeground(Color.BLACK);
    lblCustomerName.setFont(new Font("Tahoma", Font.PLAIN, 15));
    lblCustomerName.setBounds(35, 79, 189, 25);
    panelCustomerS.add(lblCustomerName);

    txtCustomerName = new JTextField();
    txtCustomerName.setEditable(true);
    txtCustomerName.setColumns(10);
    txtCustomerName.setBackground(new Color(230, 230, 250));
    txtCustomerName.setBounds(259, 79, 270, 25);
    panelCustomerS.add(txtCustomerName);

    data();
}
}

```

- **Program Name:-**ModifySupplier.java

Program Description:- This form is used to delete the Supplier info.

Code:-

```

package com.chsm.form.modify;
import java.awt.*;

```

```
import java.sql.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;

import com.chsm.form.db.DBSupplier;
import com.toedter.calendar.JDateChooser;

import DBmodel.sqlconnection;

@SuppressWarnings("serial")
public class ModifySupplier extends JFrame {

    private JPanel panel, panelSupplierpi, panelSupplieroi, panelOperation,
panelSupplierS;
    private JLabel lblSupplierName, lblSupplierRegisteredDate, lblSupplierId,
lblSupplierBalanceAmount, lblSupplierGender, lblSupplierDebitAmount,
lblSupplierCreditAmount, lblSupplierAddress, lblSupplierContact, lblSupplierEmailId;
    private JTextField txtSupplierName, txtSupplierContact, txtSupplierEmailId,
txtSupplierCrAmount, txtSupplierDeAmount, txtSupplierBAmount;
    @SuppressWarnings("rawtypes")
    private JComboBox txtSuppliedId;
    private JDateChooser txtSupplierRDate;
    private JScrollPane scrollPane;
    private JRadioButton rbMale, rbFemale;
    private ButtonGroup buttonGroup;
    final JTextArea txtSupplierAddress ;
    private JButton btnUpdate, btnClear, btnExit, btnDelete;
    private int sid;
    private DBSupplier db = new DBSupplier();

    public static java.sql.Date datemethad(java.util.Date date)
    {
        if(date != null)
        {
            java.sql.Date sqld = new Date(date.getTime());
            return sqld;
        }
        return null;
    }

    @SuppressWarnings("unchecked")
```

```
public void data()
{
    try
    {
        Connection c =sqlconnection.getConnection();
        String sql = "Select * from supplier";
        PreparedStatement pst = c.prepareStatement(sql);
        ResultSet rs = pst.executeQuery();
        if(!rs.isBeforeFirst())
        {
            JOptionPane.showMessageDialog(null, "there is no data of
Supplier in table....");
        }
        else
        {
            while(rs.next())
            {
                txtSuppliedId.addItem(rs.getInt("id"));
            }
        }
        pst.close();
    } catch (SQLException e) { e.printStackTrace(); }
}
```

```
@SuppressWarnings("rawtypes")
public ModifySupplier()
{
    setTitle("Searching_Supplier_Information");
    setBounds(100, 100, 1200, 525);
    panel = new JPanel();
    panel.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(panel);
    panel.setLayout(null);

    panelSupplierpi = new JPanel();
    panelSupplierpi.setLayout(null);
    panelSupplierpi.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Supplier_Personal_Informations", TitledBorder.LEADING, TitledBorder.TOP, null, null));
    panelSupplierpi.setBounds(22, 163, 564, 307);
    panel.add(panelSupplierpi);

    lblSupplierRegisteredDate = new JLabel("Supplier Registered Date :");
```

```
lblSupplierRegisteredDate.setForeground(Color.BLACK);  
lblSupplierRegisteredDate.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblSupplierRegisteredDate.setBounds(35, 24, 189, 25);  
panelSupplierpi.add(lblSupplierRegisteredDate);
```

```
lblSupplierGender = new JLabel("Supplier Gender :");  
lblSupplierGender.setForeground(Color.BLACK);  
lblSupplierGender.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblSupplierGender.setBounds(35, 68, 189, 25);  
panelSupplierpi.add(lblSupplierGender);
```

```
buttonGroup = new ButtonGroup();
```

```
rbMale = new JRadioButton("Male");  
rbMale.setBackground(null);  
buttonGroup.add(rbMale);  
rbMale.setHorizontalAlignment(SwingConstants.CENTER);  
rbMale.setFont(new Font("Tahoma", Font.PLAIN, 15));  
rbMale.setBounds(273, 69, 64, 23);  
panelSupplierpi.add(rbMale);  
rbMale.setActionCommand("Male");
```

```
rbFemale = new JRadioButton("Female");  
rbFemale.setBackground(null);  
buttonGroup.add(rbFemale);  
rbFemale.setHorizontalAlignment(SwingConstants.CENTER);  
rbFemale.setFont(new Font("Tahoma", Font.PLAIN, 15));  
rbFemale.setBounds(388, 69, 73, 23);  
panelSupplierpi.add(rbFemale);  
rbFemale.setActionCommand("Female");
```

```
lblSupplierAddress = new JLabel("Supplier Address :");  
lblSupplierAddress.setForeground(Color.BLACK);  
lblSupplierAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblSupplierAddress.setBounds(35, 110, 189, 25);  
panelSupplierpi.add(lblSupplierAddress);
```

```
lblSupplierContact = new JLabel("Supplier Contact :");  
lblSupplierContact.setForeground(Color.BLACK);  
lblSupplierContact.setFont(new Font("Tahoma", Font.PLAIN, 15));  
lblSupplierContact.setBounds(35, 219, 189, 25);  
panelSupplierpi.add(lblSupplierContact);
```

```
txtSupplierContact = new JTextField();
txtSupplierContact.setEditable(true);
txtSupplierContact.setColumns(10);
txtSupplierContact.setBackground(new Color(230, 230, 250));
txtSupplierContact.setBounds(259, 219, 270, 25);
panelSupplierpi.add(txtSupplierContact);

lblSupplierEmailId = new JLabel("Supplier Email ID :");
lblSupplierEmailId.setForeground(Color.BLACK);
lblSupplierEmailId.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierEmailId.setBounds(35, 263, 189, 25);
panelSupplierpi.add(lblSupplierEmailId);
```

```
txtSupplierEmailId = new JTextField();
txtSupplierEmailId.setEditable(true);
txtSupplierEmailId.setColumns(10);
txtSupplierEmailId.setBackground(new Color(230, 230, 250));
txtSupplierEmailId.setBounds(259, 263, 270, 25);
panelSupplierpi.add(txtSupplierEmailId);
```

```
scrollPane = new JScrollPane();
scrollPane.setBounds(259, 110, 270, 90);
panelSupplierpi.add(scrollPane);
```

```
txtSupplierAddress = new JTextArea();
txtSupplierAddress.setEditable(true);
txtSupplierAddress.setBackground(new Color(230, 230, 250));
scrollPane.setViewportView(txtSupplierAddress);
```

```
txtSupplierRDate = new JDateChooser();
txtSupplierRDate.setBounds(259, 24, 270, 25);
panelSupplierpi.add(txtSupplierRDate);
```

```
panelSupplieroi = new JPanel();
panelSupplieroi.setLayout(null);
panelSupplieroi.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Supplier_Official_Information", TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelSupplieroi.setBounds(608, 20, 564, 244);
panel.add(panelSupplieroi);
```

```
lblSupplierCreditAmount = new JLabel("Supplier Credit Amount :");
lblSupplierCreditAmount.setForeground(Color.BLACK);
```

```
lblSupplierCreditAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierCreditAmount.setBounds(38, 42, 189, 25);
panelSupplieroi.add(lblSupplierCreditAmount);
```

```
txtSupplierCrAmount = new JTextField();
txtSupplierCrAmount.setEditable(true);
txtSupplierCrAmount.setColumns(10);
txtSupplierCrAmount.setBackground(new Color(230, 230, 250));
txtSupplierCrAmount.setBounds(253, 42, 270, 25);
panelSupplieroi.add(txtSupplierCrAmount);
```

```
lblSupplierDebitAmount = new JLabel("Supplier Debit Amount :");
lblSupplierDebitAmount.setForeground(Color.BLACK);
lblSupplierDebitAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierDebitAmount.setBounds(38, 109, 189, 25);
panelSupplieroi.add(lblSupplierDebitAmount);
```

```
txtSupplierDeAmount = new JTextField();
txtSupplierDeAmount.setEditable(true);
txtSupplierDeAmount.setText("0");
txtSupplierDeAmount.setColumns(10);
txtSupplierDeAmount.setBackground(new Color(230, 230, 250));
txtSupplierDeAmount.setBounds(253, 109, 270, 25);
panelSupplieroi.add(txtSupplierDeAmount);
```

```
lblSupplierBalanceAmount = new JLabel("Supplier Balance Amount :");
lblSupplierBalanceAmount.setForeground(Color.BLACK);
lblSupplierBalanceAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierBalanceAmount.setBounds(38, 176, 189, 25);
panelSupplieroi.add(lblSupplierBalanceAmount);
```

```
txtSupplierBAmount = new JTextField();
txtSupplierBAmount.setEditable(false);
txtSupplierBAmount.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        int i=(Integer.parseInt(txtSupplierCrAmount.getText())-
Integer.parseInt(txtSupplierDeAmount.getText()));
        txtSupplierBAmount.setText(String.valueOf(i));
    }
});
txtSupplierBAmount.setColumns(10);
txtSupplierBAmount.setBackground(new Color(230, 230, 250));
```

```
txtSupplierBAmount.setBounds(253, 176, 270, 25);
panelSupplieroi.add(txtSupplierBAmount);

panelOperation = new JPanel();
panelOperation.setLayout(null);
panelOperation.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Operation ",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelOperation.setBounds(608, 275, 564, 195);
panel.add(panelOperation);

btnUpadate = new JButton("Update ");
btnUpadate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        int i =
Integer.parseInt((txtSuppliedId.getSelectedItem()).toString());

        db.updatesupplier(i, txtSupplierName.getText(),
datemethad(txtSupplierRDate.getDate()),
buttonGroup.getSelection().getActionCommand(), txtSupplierAddress.getText(),
txtSupplierContact.getText(),
txtSupplierEmailId.getText(), Integer.parseInt(txtSupplierCrAmount.getText()),

Integer.parseInt(txtSupplierBAmount.getText()));

JOptionPane.showMessageDialog(null, "Update Supplier Data
Sucesfull....");

        dispose();
    }
});
btnUpadate.setForeground(Color.BLACK);
btnUpadate.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnUpadate.setBackground(new Color(230, 230, 250));
btnUpadate.setBounds(88, 31, 150, 50);
panelOperation.add(btnUpadate);

btnDelete = new JButton("Delete");
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        int i =
Integer.parseInt((txtSuppliedId.getSelectedItem()).toString());
```

```
        db.dlsup(i);
        JOptionPane.showMessageDialog(null, "Supplier Data delete
Successfully....");
        dispose();
    }
});
btnDelete.setForeground(Color.BLACK);
btnDelete.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnDelete.setBackground(new Color(230, 230, 250));
btnDelete.setBounds(88, 112, 150, 50);
panelOperation.add(btnDelete);

btnClear = new JButton("Clear");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        txtSupplierAddress.setText(null);
        txtSupplierName.setText(null);
        txtSupplierContact.setText(null);
        txtSupplierEmailId.setText(null);
        txtSupplierCrAmount.setText(null);
        txtSupplierDeAmount.setText("0");
        txtSupplierBAmount.setText(null);
        txtSupplierRDate.setDate(null);
        buttonGroup.clearSelection();
    }
});
btnClear.setForeground(Color.BLACK);
btnClear.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnClear.setBackground(new Color(230, 230, 250));
btnClear.setBounds(326, 31, 150, 50);
panelOperation.add(btnClear);

btnExit = new JButton("Exit");
btnExit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        JOptionPane.showMessageDialog(null, "Do you want to close
Modify_Supplier....");
        dispose();
    }
});
btnExit.setForeground(Color.BLACK);
btnExit.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnExit.setBackground(new Color(230, 230, 250));
```

```
        btnExit.setBounds(326, 112, 150, 50);
        panelOperation.add(btnExit);

        panelSupplierS = new JPanel();
        panelSupplierS.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Supplier_Search",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
        panelSupplierS.setBounds(22, 20, 564, 132);
        panel.add(panelSupplierS);
        panelSupplierS.setLayout(null);

        lblSupplierId = new JLabel("Supplier Id :");
        lblSupplierId.setForeground(Color.BLACK);
        lblSupplierId.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblSupplierId.setBounds(35, 27, 189, 25);
        panelSupplierS.add(lblSupplierId);

        lblSupplierName = new JLabel();
        lblSupplierName.setText("Supplier Name :");
        lblSupplierName.setForeground(Color.BLACK);
        lblSupplierName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblSupplierName.setBounds(35, 79, 189, 25);
        panelSupplierS.add(lblSupplierName);

        txtSupplierName = new JTextField();
        txtSupplierName.setEditable(true);
        txtSupplierName.setBackground(new Color(230, 230, 250));
        txtSupplierName.setBounds(259, 79, 270, 25);
        panelSupplierS.add(txtSupplierName);
        txtSupplierName.setColumns(10);

        txtSuppliedId = new JComboBox();
        txtSuppliedId.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {

sid=Integer.parseInt(txtSuppliedId.getSelectedItem().toString());
                try
                {
                    final Connection c =sqlconnection.getConnection();
                    final String sql = "Select * from supplier where id=?";
                    final PreparedStatement pst = c.prepareStatement(sql);
                    pst.setInt(1, sid);
```

```
        final ResultSet rs = pst.executeQuery();
        while(rs.next())
        {
            txtSupplierName.setText(rs.getString("name"));

            txtSupplierRDate.setDate(rs.getDate("reg_date"));

            txtSupplierContact.setText(rs.getString("contact"));

            txtSupplierAddress.setText(rs.getString("address"));

            txtSupplierContact.setText(rs.getString("contact"));

            txtSupplierEmailId.setText(rs.getString("email"));

            txtSupplierCrAmount.setText(rs.getString("balance"));
            txtSupplierDeAmount.setText("0");
            final String gender1=rs.getString("gender");
            if(gender1.equals("Male"))
            {
                rbMale.setSelected(true);
            }
            else
            {
                rbFemale.setSelected(true);
            }
        }
    }
    catch(final Exception ex)
    {
        ex.printStackTrace();
    }
    });
    txtSuppliedId.setBounds(259, 27, 270, 25);
    panelSupplierS.add(txtSuppliedId);

    data();
}
}
```

- **Program Name:-**DBProduct.java

Program Description:- This form is used to save and modify the product info.

Code:-

```
package com.chsm.form.db;
import java.awt.*;
import java.sql.*;

public class DBProduct
{

    private String query1, query2;
    private Connection con;
    private PreparedStatement pst;
    private ResultSet rs;
    private String id=null;

    public DBProduct()
    {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/chsm", "root", "root");
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }

    public String setID()
    {
        try
        {
            query1 = "select id from product";
            pst = con.prepareStatement(query1);
            rs = pst.executeQuery();
            if(!rs.isBeforeFirst())
                id="1";
            else
            {
                rs.last();
                id = String.valueOf(rs.getInt(1) + 1);
            }
        }
        catch (HeadlessException | SQLException e)
```



```

        {
            e.printStackTrace();
        }

        return id;
    }

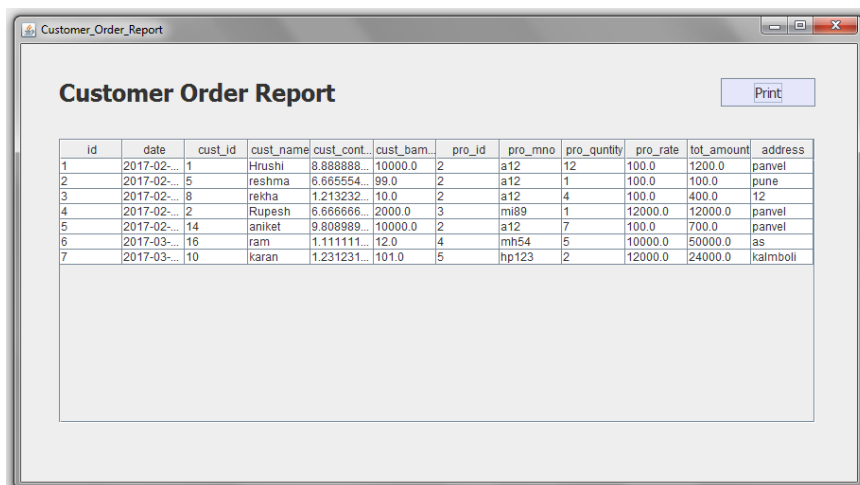
    public void addproduct(int id, String c_name, String p_name, String model_no, String
color, String description, int prise)
    {
        try {
            query2="insert into product value(?,?,?,?);";
            pst=con.prepareStatement(query2);

            pst.setInt(1, id);
            pst.setString(2, c_name);
            pst.setString(3, p_name);
            pst.setString(4, model_no);
            pst.setString(5, color);
            pst.setString(6, description);
            pst.setInt(7, prise);
            pst.executeUpdate();

        } catch (HeadlessException | SQLException e) {
            e.printStackTrace();
        }
    }
}

```

Program Description:- This form is used to view Customer Order Report



id	date	cust_id	cust_name	cust_cont	cust_bam	pro_id	pro_mno	pro_quntity	pro_rate	tot_amount	address
1	2017-02-...	1	Hrushu	8.888888...	10000.0	2	a12	12	100.0	1200.0	panvel
2	2017-02-...	5	reshma	6.665554...	99.0	2	a12	1	100.0	100.0	pune
3	2017-02-...	8	rekha	1.213232...	10.0	2	a12	4	100.0	400.0	12
4	2017-02-...	2	Rupesh	6.666666...	2000.0	3	mi89	1	12000.0	12000.0	panvel
5	2017-02-...	14	aniket	9.808989...	10000.0	2	a12	7	100.0	700.0	panvel
6	2017-03-...	16	ram	1.111111...	12.0	4	mh54	5	10000.0	50000.0	as
7	2017-03-...	10	karan	1.231231...	101.0	5	hp123	2	12000.0	24000.0	kalmoli

Code:-

```
package com.chsm.form.report;
import java.sql.*;
import java.text.*;
import javax.swing.*;
import javax.swing.border.*;
import net.proteanit.sql.*;
import java.awt.*;
import java.awt.event.*;
import DBmodel.sqlconnection;

@SuppressWarnings("serial")
public class RpdCustomerOrder extends JFrame {

    private JPanel contentPane;
    private JTable table;
    private JLabel lblNewLabel;
    private JButton button;

    void view ()
    {
        try
        {
            Connection c=sqlconnection.getConnection();
            String s="select * from cust_order";
            PreparedStatement pst=c.prepareStatement(s);
            ResultSet rs=pst.executeQuery();
            table.setModel(DbUtils.resultSetToTableModel(rs));
        }

        catch(Exception e)
        {
            e.printStackTrace();
        }
    }

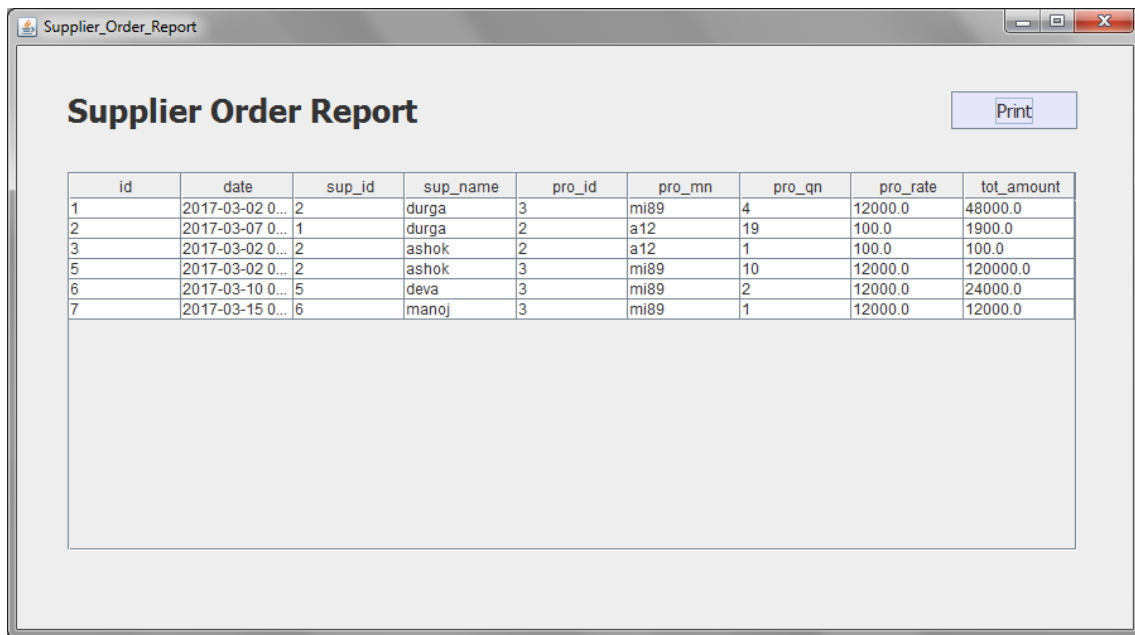
    public RpdCustomerOrder() {
        setTitle("Customer_Order_Report");
        setBounds(100, 100, 900, 500);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
    }
}
```

```
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(40, 100, 800, 300);
contentPane.add(scrollPane);
table = new JTable();
scrollPane.setViewportView(table);
lblNewLabel = new JLabel("Customer Order Report");
lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 25));
lblNewLabel.setBounds(40, 36, 400, 30);
contentPane.add(lblNewLabel);
button = new JButton("Print");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        MessageFormat header = new MessageFormat("Report print");
        MessageFormat footer = new
MessageFormat("Page{0,number,integer}");
        try{
            table .print(JTable.PrintMode.NORMAL,header,footer);
        } catch (java.awt.print.PrinterException ex){
            System.err.format("Can not print %s %n",ex.getMessage());
        }
    }
});
button.setFont(new Font("Tahoma", Font.PLAIN, 15));
button.setBackground(new Color(230, 230, 250));
button.setBounds(740, 36, 100, 30);
contentPane.add(button);

view();
}
}
```

- **Program Name:-** RPDSuppOrder.java

Program Description:- This form is used for Show Supplier Order Report.



Code:-

```
package com.chsm.form.report;
import java.sql.*;
import java.text.*;
import javax.swing.*;
import javax.swing.border.*;
import net.proteanit.sql.*;
import java.awt.*;
import java.awt.event.*;
import DBmodel.sqlconnection;

@SuppressWarnings("serial")
public class RpdSupplierOrder extends JFrame {

    private JPanel contentPane;
    private JTable table;
    private JButton button;
    private JLabel lblSupplierOrderReport;

    void view()
    {
        try
        {
```

```
        Connection c=sqlconnection.getConnection();
        String s="select * from sup_order";
        PreparedStatement pst=c.prepareStatement(s);
        ResultSet rs=pst.executeQuery();
        table.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}

public RpdSupplierOrder() {
    setTitle("Supplier_Order_Report");
    setBounds(100, 100, 900, 500);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setBounds(40, 100, 800, 300);
    contentPane.add(scrollPane);

    table = new JTable();
    scrollPane.setViewportView(table);

    button = new JButton("Print");
    button.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            MessageFormat header = new MessageFormat("Report print");
            MessageFormat footer = new
MessageFormat("Page{0,number,integer}");

            try{
                table .print(JTable.PrintMode.NORMAL,header,footer);
            } catch (java.awt.print.PrinterException ex){
                System.err.format("Can not print %s %n",ex.getMessage());
            }
        }
    });
    button.setFont(new Font("Tahoma", Font.PLAIN, 15));
    button.setBackground(new Color(230, 230, 250));
```

```
        button.setBounds(740, 36, 100, 30);
        contentPane.add(button);

        lblSupplierOrderReport = new JLabel("Supplier Order Report");
        lblSupplierOrderReport.setFont(new Font("Tahoma", Font.BOLD, 25));
        lblSupplierOrderReport.setBounds(40, 36, 400, 30);
        contentPane.add(lblSupplierOrderReport);
        view();
    }
}
```

CHAPTER-7

TESTING

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect-free in order to produce a quality product.

7.1 Testing Approaches

There are three types of software testing approaches.

1. White Box Testing
2. Black Box Testing

7.1.1 White Box Testing

It is also called Glass Box, Clear Box, Structural Testing. White Box Testing is based on the application's internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing is usually done at the unit level.

7.1.2 Black Box Testing

It is also called Behavioral/Specification-Based/Input-Output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.

7.2 Unit Testing

Unit Testing is done to check whether the individual modules of the source code are working properly. i.e. testing each and every unit of the application separately by the developer in the developer's environment. It is AKA Module Testing or Component Testing.

7.3 Integration Testing

Integration Testing is the process of testing the connectivity or data transfer between a couple of unit tested modules. It is AKA I&T Testing or String Testing. It is subdivided into the Top-Down Approach, Bottom-Up Approach, and Sandwich Approach (Combination of Top-Down and Bottom-Up).

7.4 System Testing (End to End Testing)

It's a black box testing. Testing the fully integrated application is also called as an end to end scenario testing. To ensure that the software works in all intended target systems. Verify thorough testing of every input in the application to check for desired outputs. Testing of the user's experiences with the application.

7.5 Acceptance Testing

To obtain customer sign-off so that software can be delivered and payments received. Types of Acceptance Testing are Alpha, Beta & Gamma Testing.

CHAPTER-8

SYSTEM IMPLEMENTATION

The steps for making Class file of the project is as follows: -

- Open your project.
- Compile the Java file.
- Then class file will be created.
- Execute that Java Class file.

CHAPTER-9

FUTURE ENHANCEMENT

In my future version of this system I would like to add some following features in my project:

1. Some more reports will be added to make administrator task easier.
2. Interface will be made more user friendly.

Being a computer system, the system has lots of scope. It not only carried out work faster but also efficiently. A lots of manual work like searching ,personal alias is reduced to a large extent. As this is computerized, system the manager is completely depends on computer for accessing details about students. Here incase there is power failure or some hardware problem which cannot be fixed easily, the manager cannot continue this work. He gets struck in the middle of the day and this dealing need to be postponed.

Also the software need to be maintained properly from time to time, that is if it requires being updated or modified etc. then the manager need to take care of it.

CHAPTER-10

CONCLUSION

10.1 Conclusion of the COMPUTER HARDWARE SHOP MANAGEMENT SYSTEM

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objectives of software planning is to provide a frame work that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

10.2 At the end it is concluded that we have made effort on following points...

- ❖ A description of the background and context of the project and its relation to work already done in the area.
- ❖ Made statement of the aims and objectives of the project.
- ❖ The description of Purpose, Scope, and applicability.
- ❖ We define the problem on which we are working in the project.
- ❖ We describe the requirement Specifications of the system and the actions that can be done these things.
- ❖ We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- ❖ We included features and operations in detail, including screen layouts.
- ❖ We designed user interface and security issues related to system.
- ❖ Finally the system is implemented and tested according to test cases.

CHAPTER-11

BIBLIOGRAPHY

11.1 BIBLIOGRAPHICAL BOOKS:

1. Java Programming: Advanced Topics by Joe Wigglesworth and Paula McMillan, Thomson Course, Third Edition 2004
2. Java 2: The Complete Reference by Herbert Schildt, Tata McGraw Hill, Fifth Edition
3. Web Enabled Commercial Applications Development Using Java 2 by Ivan Bayross, BPB Publication, Revised Edition 2006

11.2 BIBLIOGRAPHICAL SITES :

- <http://www.google.co.in/>
- <http://sun.java.com/>
- <http://www.yahoo.co.in/>
- <http://www.w3school.com/>
- <http://www.syncjedit.co.in/>